潜在的情報を利用した文識別モデル

岡野原 大輔† 辻井 潤一†‡§
†東京大学情報理工学系研究科コンピュータ科学専攻
‡School of Computer Science, University of Manchester
§NaCTeM (National Center for Text Mining)
{ hillbig, tsujii }@is.s.u-tokyo.ac.jp

1 はじめに

言語モデルは与えられた文が文法的，用法的に正しいかどうかを判定するために使われ，音声認識，機械翻訳など多くのアプリケーションで利用されている．例えば，確率的言語モデルは単語列の生成確率を求め，確率が大きいものを尤もらしい単語列として選ぶ．

その中でも特にN-gramモデルが単純ながら，強力であるため広く利用されている．N-gramモデルは文 $S := w_1^t$ の生成確率を $P(S) = \prod_{i=1...t} P(w_i|w_{i-N+1}^{i-1})$ として求める．これは各単語が直前 $N-1$ 単語にのみ条件付けされて生成される条件付確率の結合確率である．

しかし，N-gramモデルが与える確率は文長に強く依存し，各単語のコーパス中での頻度にも依存する．例えば，次の二文 S_1 :=“I are fine.”，S_2 :=“I am fine and getting better.” では，S_2 が文法的に正しいにも関わらず，文長が長いため，$P(S_1) > P(S_2)$ となり，S_3 :=“I live in Tokyo.” と S_4 :=“I live in Hongo” では，どちらの文も正しいが “Hongo” の頻度が “Tokyo” より少ないため，$P(S_3) > P(S_4)$ となる．

これらの問題は今まで顕著化してこなかった．なぜなら，機械翻訳や音声認識など多くのアプリケーションにおいて，言語モデルは，文長がほぼ同じで，出現単語の傾向も似ている複数の文から尤もらしい文を選ぶタスクに利用されていたからである．しかし，N-gramモデルを利用して文の相対的ではなく絶対的な正しさを測ることは，困難である．例えば，英語を母語としない人により書かれた英文の中から間違った英文を発見するタスクにおいて，N-gramモデルによってあり得ない単語列を発見することは可能だが，文の正しさを測ることや，動詞が一つも含まれない文のように，局所的に正しくても全体で間違っている文を発見することは難しい．

本稿では，文が正しいかどうかを直接識別する文識別モデルを提案する．この実現のためには “正しくない” 文をどのように手に入れるか，大量の訓練数，特徴種類数を扱える識別器をどのように構築するかを解決する必要がある．本稿では確率的言語モデルからのサンプリングによって “正しくない” 文を生成する．また，N-gramを特徴として直接扱うのではなく，それらのクラスタリングした結果を Semi-Markov Class Model によって求め，特徴として利用することを提案する．

実験結果において，サンプリングによって得られた負例は実際の負例とみなせること，また，提案学習方法が約百万文といった大量の訓練データから学習可能であり，文を正しく識別可能であることを示す．

2 文識別モデル

文識別モデルは各文 S に対し，文の正しさを評価するスコア $f(S)$ を与え，$f(S) > 0$ ならば文は正しい，$f(S) < 0$ ならば文は間違っていると判定する．例えば，確率的言語モデルも $f(S) = P(S)/|S| - c$ のように確率を文長 $|S|$ で正規化し，その値が基準値 c より大きいかによって文識別モデルとしてみなすことができる．

本稿では文識別モデルを線形モデルとして表す．文 S から得られる I(S は “I am” を含む) や I(S は動詞を含む) といった各特徴の成分値を並べた特徴ベクトルを $\phi(S)$ とする．この特徴設計には任意の計算可能な特徴を利用することが可能であり，N-gramモデルでは難しかった重複のある特徴や遠距離情報を利用可能である．文識別モデル $f(S)$ はモデルのパラメータである重みベクトル $\mathbf{w}$ を用いて

$$f(S) = \mathbf{w} \cdot \phi(S) \tag{1}$$

として表される．$\mathbf{w}$ は訓練データを用いて推定される．

こうした文識別モデルは以前より提案されていたが，限られた候補文からのリランキング [1] や，かな漢字変換タスク [2] のような入力とその正解単語列が与えられる場面のみで利用され，確率的言語モデルのようにあらゆる文に対して利用することはできなかった．その主な理由として N-gram モデルは大量の生コーパスを訓練データとして利用できるのに対し，文識別モデルは正解付きの訓練データが必要ということが挙げられる．つまり，“正しい” 文は大量に入手できるのに対し “正しくない” 文を大量に入手できないという問題点があった．我々の提案する文識別モデルの枠組みは従来の N-gram 同様，大量の生コーパスを訓練データとして利用できる．

```
For i=1,2,...
   単語 w_i を P(w_i|w_{i-N+1}^{i-1}) に従ってサンプリングする.
   If w_i = "文の終端" then break;
End
```

図 1: N-gram モデルからサンプリングするコード例

```
We know of no program, and animated
discussions about prospects for trade
barriers or regulations on the rules of
the game as a whole, and elements of
decoration of this peanut-shaped to
priorities tasks across both target countries
```

図 2: Tri-gram モデルからサンプルされた文の例

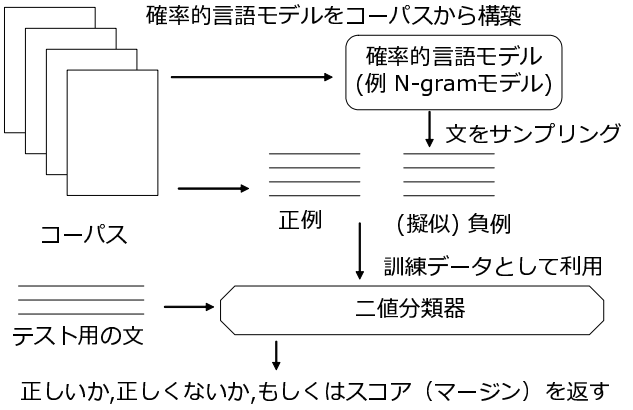

図 3: 提案手法の概略図

3 擬似負例を利用した文識別モデル

本稿は，識別器の訓練のために “正しくない” 文（以降，これを負例と呼ぶ）を確率的言語モデルから生成することを提案する．ランダムな単語列を生成することで，負例を得ることもできるが，バリエーションが非常に大きく，そのため学習に必要なサンプル数は非常に大きくなり，非現実的である．しかし，殆どのランダムな単語列である負例は N-gram の簡単なチェックで識別可能であり，また，アプリケーションにおいても，識別対象の文は正しい文に似ている場合が多い．

これらをふまえ，本稿では確率的言語モデルから，その確率に従って負例をサンプルする．これらは実際に負例であるかは分からないため，以降，このサンプルされた文を擬似負例と呼ぶ．図 3 に提案手法の概略図を示す．提案手法では N-gram モデルなど簡単なモデルでは識別できない負例のみ識別するよう学習する．本稿では Tri-gram モデルに線形補間スムージングを適用した確率分布から文をサンプリングした．図 1 に擬似負例を N-gram からサンプリングするコード例を示す．図 2 が実際にサンプリングされた文の例である．擬似負例の特徴は，局所的には正しいが，文全体では肯定的な表現と否定的な表現が混ざるなど整合性が無いことである．

4 大規模データに対応した学習

提案手法の実現には大量の訓練データを用いて学習を行う必要があり，学習の計算量が非常に大きくなる．しかし，従来の文識別モデルでは訓練データ数が限られ，また組み合わせ特徴（カーネルトリック）も利用されなかったため，計算コストが少なくこの問題は起こらなかった．本稿では，訓練データをまとめて処理するバッチ学習では計算コストが大きいことからオンライン最大マージン学習 [3] を利用する．

はじめに，初期重みベクトル $\mathbf{w}_1$ を 0 に初期化する．次に，各訓練例 $\mathbf{x}_i := \phi(S_i)$ を順に観察し，各ラベルが $+1$ か -1 であるかを予測した後，正解ラベル y_i と比較し hinge-loss $l(\mathbf{w}; (\mathbf{x}_i, y_i)) = \max(0, 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t))$ を求める．そして分類器が予測を間違えた場合，または予測の信頼度が低い場合に重みベクトルを以下を満たすように更新する，

$$\mathbf{w}_{t+1} = argmin_{\mathbf{w}} \frac{1}{2}||\mathbf{w} - \mathbf{w_t}||^2 + C\xi \tag{2}$$

$$\text{subject to } l(\mathbf{w}; (\mathbf{x}_t, y_t)) \leq \xi \text{ and } \xi \geq 0, \tag{3}$$

ただし ξ はスラック変数であり，C は ξ の目的関数への影響力を調整する変数である．C が大きいならばより積極的が更新が行われる．この更新式は解析的に次のようにして求められる [3],

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t. \tag{4}$$

ただし $\tau_t = \min\{C, \frac{l_t}{\|\mathbf{x}_t\|^2}\}$ である．SVMs と同様に，最終重みベクトルは $\mathbf{w} \cdot \mathbf{x} = \sum_i \tau_t y_i \langle \mathbf{x_i} \cdot \mathbf{x} \rangle$ のように訓練サンプルのカーネル線形和としてあらわされ，内積部分は任意のカーネル $K(\mathbf{x}_i, \mathbf{x}_t)$ に置き換えることができる．

文識別モデルにおいて特徴関数の組み合わせは特に重要である．カーネルトリックをオンライン学習に適用した場合，SVMs における support set と同様に，観察した訓練サンプルの部分集合である active set が保存される．しかし support set とは違い，学習器が予測を間違える度に訓練サンプルが active set に追加されるため，active set の数は非常に大きくなる．カーネルトリックを用いたとしても計算量は無視できない．

Kernel 計算の高速化のために工藤らによって提案された PKI を利用する [4]．これは情報検索における転置ファイルと同様に，各特徴がどの訓練データにおいて発火したかを記録したものである．$f(\mathbf{x})$ を求めるときは，$\mathbf{x}$ で発火している各特徴について，今までに発火した訓練データを列挙し，各訓練データでの内積値に加算する．次に,, それらの内積値を用いて $\mathbf{x}$ と各訓練データとのカーネル値を求める．

PKI はオンライン学習の場合でもインデクス末尾への追加操作が加わるのみであり，逐次更新が可能である．

5 潜在的特徴の抽出

提案手法のもう一つの問題点は，全ての N-gram が特徴として使われる可能性があるため，特徴種類数が非常に大きいことである．カーネルトリックを使用する場合，特徴ベクトルの集合からなる active set を保持する必要もあり，メモリ使用量が特に問題となる．この問題を解決するためには影響力の小さい特徴を除外していくことも考えられるが，オンライン学習中において，どの特徴の影響力が小さいかを判断するのは困難である．本稿では N-gram を Semi-Markov Class Model(SMCM) によってクラスタリングし，その結果を特徴として利用することを提案する．SMCM は本稿で初めて提案する．

Class Model(CM) は元々言語モデルとして提案された [5]．CM は各単語から単語種類数よりはるかに少ない数のクラスへのマッピングを求める．SMCM は，各単語ではなく，可変長の各チャンク（N-gram) をクラスにマッピングしたものである．SMCM においては各文は可変長のチャンク列に分割される．これらは全て訓練コーパスを用いて自動的に求められる．Bi-gram CM，Bi-gram SMCM において，訓練コーパス w_1^t の対数尤度はそれぞれ次のように求められる

$$P_{cm}(w_1^t) = \prod_i P(w_{i+1}|c_{i+1})P(c_{i+1}|c_i), \quad (5)$$

$$P_{smcm}(w_1^t) = \sum_s \prod_i P(c_i|c_{i-1})P(w_{s(i)}^{e(i)}|c_i). \quad (6)$$

但し s は S の可能な全ての分割であり，$s(i)$ は i 番目のチャンクの開始位置，$e(i)$ は終了位置であり，$s(i+1) = e(i)+1$ が成り立つ．この対数尤度を最大化するようなクラス割り当てを求めることによって各単語の割り当てを求める．チャンクの分割は対数尤度を最大化する Viterbi 分割の結果を利用し，再帰的に分割とクラスタリングを繰り返した．

また，SMCMs は計算量が非常に大きいので，CMs へのサンプリングおよびボトムアップクラスタリングによる構築を用いた高速化 [6] を同様に適用した．

6 実験

実験データとして BNC-corpus を用いた．はじめに BNC-corpus を，*model-train*（450 万文），*DLM-train-positive*（25 万文），*DLM-test-positive*（5 千文）に分割した．次に N-gram モデルを *model-train* を用いて構築した．そこから，擬似負例を 25 万文，5 千文ずつサンプリングし，それぞれを *DLM-trian-positive*，*DLM-test-positive* とシャッフルし *DLM-train*，*DLM-test* を作った．単語数が 5 以下の文は複合語である可能性もあり識別が難しいため，前もって取り除いた．SMCMs は *model-train* を用いてクラス数が 100 と 500 の二つを構築した．どちらの SMCM においても抽出されたチャンク数は約 280 万であった．

表 1: テストデータにおける結果

	精度 (%)	訓練時間 (s)
Linear classifier		
単語 tri-gram	51.28	137.1
POS tri-gram	52.64	85.0
SMCM (クラス数 100)	51.79	304.9
SMCM (クラス数 500)	54.45	422.1
3-order Polynomial Kernel		
単語 tri-gram	73.65	20143.7
品詞 tri-gram	66.58	29622.9
SMCM (クラス数 100)	67.11	37181.6
SMCM (クラス数 500)	74.11	34474.7

はじめに擬似負例に対する予備実験を行った．英語を母語とする被験者に，*DLM-train* からサンプルされた 100 文に対し，その文が正しいか，間違っているかを判定してもらった．結果は全ての正例は正しいと判断され，負例は一例を除き他は全て間違っていると判断された[1]．この結果から擬似負例は負例としてみなしてよいことが言える．また，採点者が一文を判定するのに平均 25 秒かかっており，判定は人にも難しいことがいえる．

次に，これらの文が構文解析器によって識別できるかどうかを調べた．Charniak Parser[7]，HPSG Parser[8] の二種類の構文解析を利用し，*DLM-train* からサンプルされた 100 文を解析した．結果は正例の一つを除き全て正常に構文解析が終了した．この結果から擬似負例は構文的特徴の違いで識別できないことが言える．

次に，提案識別器を用いた実験を行った．はじめに各特徴，及びカーネルを用いたときの性能比較を行った．用いた特徴は単語の Tri-gram，品詞情報[2]の Tri-gram，及び SMCM の結果として求められたクラスの Bi-gram である．*DLM-train* を訓練用データとして，*DLM-test* を評価用データとして用いた．全ての実験において $C = 50$ を用いた．また，カーネル計算には PKI を用いた．

表 1 に各特徴，及び 3 次多項式カーネルを用いたときの結果を示す．この結果からカーネルが識別に重要であること，また SMCM が N-gram と同程度の性能であることが分かる．表 2 に，各特徴の実際使われた種類数を示す．実際に識別器を利用する場合，これらの数に比例する作業領域量が必要になる．これらの実験結果から SMCMs の特徴種類数は N-gram と比較し非常に少ないながら，同程度の性能を達成できていることが分かる．なお，訓練時間については SMCMs の場合が特に大きくなっているが，これはチャンク列から各クラスへのマッピングを求める部分の実装がナイーブであり，適切なデータ構造を使えば速くなると考えられる．

[1] 日本語を母語とする被験者二人の精度は 65%, 70% だった．

[2] 品詞付けには Genia Tagger（http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/）を用いた

表 2: 各特徴の識別に使われた種類数

	特徴種類数
単語 tri-gram	15773230
品詞 tri-gram	35376
SMCM ($\|C\| = 100$)	9335
SMCM ($\|C\| = 500$)	199745

表 3: PKI を使用した場合の計算時間の変化

	訓練時 (秒)	評価時 (ミリ秒)
ベースライン	37665.5	370.6
+ PKI	4664.9	47.8

次に，PKI を用いた結果を表 3 に示す．この実験では全てを利用した場合の計算時間が非常に大きかったため，*DLM-train* 中の 20 万文のみを用いた．この結果から PKI を用いることで訓練時，評価時ともに約 10 倍の高速化が達成できていることが分かる．

図 4 に，評価データのマージンの分布を示す．多くの文が 0 に近い一方，正例，負例はそれぞれマージンが 0 以上，0 以下側に分布しているため，識別器の判断基準を 0 以外に変更することにで，再現率もしくは精度を重視することができる．

最後に図 5 に訓練データ数に対する学習曲線を示す．この実験では SMCM bi-gram を特徴として用いた．この結果から訓練データ数をさらに増やすことにより，精度はより向上すると考えられる．訓練データはいくらでも入手できるために，識別器のスケーラビリティをさらに挙げることで，より高い精度が挙げられると考えられる．

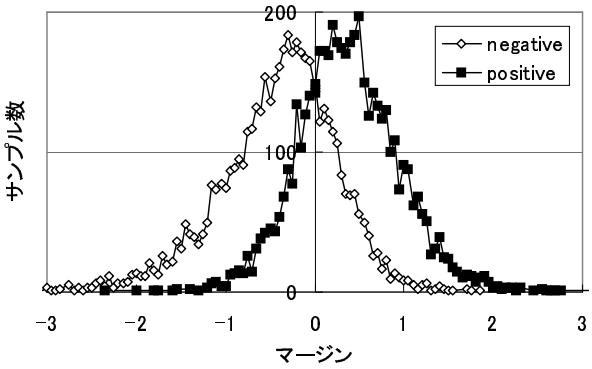

図 4: SMCM bi-gram を用いたときのマージン分布

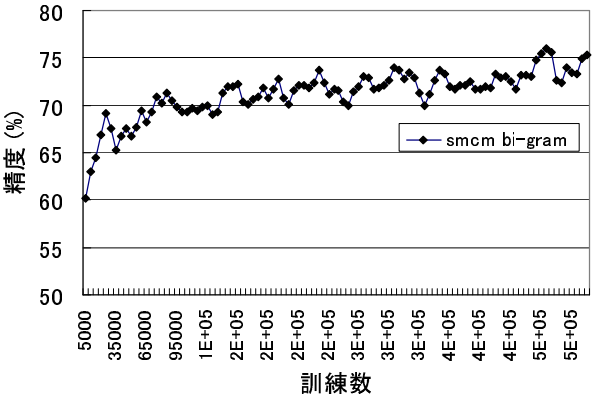

図 5: SMCM ($\|C\| = 500$) を用いたときの学習曲線

7 まとめ

本稿では，文の正しさを直接識別するモデルを提案し，その実現のため，確率的言語モデルから正しくない文を生成し，それを負例として用いることを提案した．これにより生コーパスのみから文識別モデルを学習でき，文識別モデルも，確率的言語モデルと同様の大量の訓練データを利用可能となる．

また，実験ではオンライン最大マージン学習と PKI を用いることで，N-gram モデルと同規模の大量の訓練データを使った学習が可能であることを示した．それに加え，Semi-Markov Class Model を利用することで，少数の潜在的な特徴を用いても単語列など大量の特徴を利用した場合と同じ性能を達成できること示した．

今後は，文識別に有効な特徴を調べ，実際に言語モデルを利用したアプリケーションの開発をする予定である．

また，確率分布を保持するモデルを直接操作することによりサンプルを生成せずに識別モデルを学習する手法を開発する予定である．

参考文献

[1] Brian Roark, Murat Saraclar, and Michael Collins. Discriminative n-gram language modeling. computer speech and language. *Computer Speech and Language*, 21(2):373–392, 2007.

[2] Jianfeng Gao, Hao Yu, Wei Yuan, and Peng Xu. Minimum sample risk methods for language modeling. In *Proc. of HLT/EMNLP*, 2005.

[3] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 2006.

[4] Taku Kudo and Yuji Matsumoto. Fast methods for kernel-based text analysis. In *ACL*, 2003.

[5] S. Martin, J. Liermann, and H. Ney. Algorithms for bigram and trigram word clustering. *Speech Communicatoin*, 24(1):19–37, 1998.

[6] 岡野原 大輔. Classmodel を用いた単語分類の拡張及び高速化. In **自然言語処理研究会** *(SIGNL-163)*, 2004.

[7] Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proc. of ACL*, pages 173–180, June 2005.

[8] Yusuke Miyao and Jun'ichi Tsujii. Probabilistic disambiguation models for wide-coverage hpsg parsing. In *Proceedings of ACL 2005.*, pages 83–90, Ann Arbor, Michigan, June 2005.